

Kimmo Lemetti

# Kingdom Keeper -pelin grafiikan toteutus

Assetien luonti konseptista lopulliseen malliin

Metropolia Ammattikorkeakoulu

Medianomi AMK

Viestintä

Opinnäytetyö

23.5.2016

Tekijä(t) Otsikko	Kimmo Lemetti Kingdom Keeper pelin grafiikan toteutus
Sivumäärä Aika	25 sivua 23.5.2016
Tutkinto	Medianomi AMK
Koulutusohjelma	Viestinnän koulutusohjelma
Suuntautumisvaihtoehto	3D Animaatio ja Visualisointi
Ohjaaja(t)	Lehtori Pasi Kaarto
<p>Tässä lopputyössä käydään läpi Kingdom Keeper -pelin taidesuuntauksen luominen. Tavoitteena on löytää paras tapa toteuttaa pelin visualinen osuus niin, että sen voi tehdä pieni tiimi. Teknisien ja aika rajoitusten tasapainottelu nousee tärkeäksi pohdintakohteeksi, kun pelin tuotantotiimi on vain kaksi henkilöä. Muita samankaltaisia pelejä tutkimalla määritellään sopivimmat toteutustavat ja rajoitteet taidesuuntauksen luonnille.</p> <p>Työ painottuu suurimmaksi osaksi konseptointityöhön ja suunnittelutyön tutkimiseen. Suunnitteluosuus tutkitaan alusta loppuun yhden pelissäkäytettävän tornin, luomisprosessin läpi käymisellä. Työn aikana pohditaan, mitä kaikkea tulisi ottaa huomioon ja miten luoda hyvän näköinen kokonaisuus.</p> <p>Aluksi pohditaan, mitä rajoituksia pieni tuotantotiimi antaa projektille ja tutkitaan mahdollisia toteutusmahdollisuuksia. Samantyyllisiä pelejä analysoimalla käydään läpi muutama vaihtoehto, ja lopuksi valitaan sopiva piirrettyä taidetyyli.</p> <p>Tämän jälkeen tehdään konsepti taidetta referenssejä ja moodboardia avuksi käyttäen. Ensin luodaan yleiskuvia suuntaa antamaan ja sitten tarkempia konsepteja ja kaavakuvia mallinnustyötä avustamaan.</p> <p>Lopuksi konseptien pohjalta luodaan lopullinen 3D-malli pelissä käytettäväksi. Tämä osuuden aikana tuon esiin hyviä työtapoja ja mahdollisia ongelmia. Kerron myös teksturointi prosessista ja lopullisen mallin pelimoottoriin viemisestä.</p> <p>Lopputuloksena on pelin prototyypille hyvä perustaidepaketti, josta pelin kehitystä voidaan jatkaa eteenpäin.</p>	
Avainsanat	pelejä, grafiikka, taidesuunnittelu, konseptointi

Author(s) Title	Kimmo Lemetti Graphics for Kingdom Keeper -game
Number of Pages Date	25 pages 23 May 2016
Degree	Bachelor of Media
Degree Programme	Media Communication
Specialisation option	3D Animation and Visualization
Instructor(s)	Lecturer Pasi Kaarto
<p>This thesis describes the creation of the art direction for a game called Kingdom Keeper. The goal is to find the best way to design the visual portion of the game in such a way that it can be realized by a small team. Balancing technical and time restrictions becomes an important part of concern as the team only has two members. The best techniques and possible limitations are found by analyzing other similar games.</p> <p>The work focuses mostly on the concept design and planning stages. The entire design process is explored from start to finish by creating a tower to be used in the game. The work focuses on what to keep in mind and how to create a good looking art direction.</p> <p>First we figure out what limitations a small production team brings and what possible ways there are to approach the project. By analyzing similar games, we come up with a few possibilities and then choose a fitting cartoonish art direction.</p> <p>After this we create concept art by utilizing references and moodboards. First we create overall concept pictures to give direction, and then more detailed concepts and diagrams to help with the modeling work.</p> <p>In the end a finished 3D model is created from the concept art to be used in the game. During this portion I bright up good work practices and possible problems. I also talk about the texturing process and getting the model into the game engine.</p> <p>The final product is an art base for the prototype of the game, from which the project can proceed.</p>	
Keywords	game, graphics, art direction, concept art

## Sisällys

1	Johdanto	1
2	Käsitteet	2
3	Pohja Kingdom Keeper –pelin suunnittelulle	2
3.1	Lähtökohta	2
3.2	2D vai 3D	2
3.3	Kamera	4
3.4	Vastaavien toteutuksien tutkiminen	5
4	Pelin grafiikan suunnitteluprosessi	7
4.1	Lähtökohta	7
4.2	Ideoiden haku, referenssien keräys, moodboard	8
4.3	Yleiskuvien teko	9
4.4	Ideoiden hiominen ja yksittäiset konseptit	10
4.5	Mockup ja user interface	12
5	Grafiikan tekninen toteutus	13
5.1	Teknisen työn aloitus	13
5.2	Assetien luonti	15
5.3	Mallinnus	16
5.4	Teksturointi	18
5.5	Hahmot	20
6	Assetien viimeistely ja testaus	21
6.1	Export ja Import	21
6.2	Assetien tarkistus	22
7	Yhteenveto	23
	Lähteet	24
	Kuvalähteet	25

## 1 Johdanto

Peliä voidaan kutsua indiepeliksi silloin, kun pelin luo yleisesti pieni ryhmä (joskus jopa vain yksi henkilö) ja varsinaista peliyhtiötä ei ole olemassa. Suuremman luokan peleihin verrattuna indiepelit ovat yleisesti pienempiä ja yksinkertaisempia, ja täten sopivat hyvin mobiili- ja tablet-laitteille. Näiden laitteiden yleistymisen myötä on myös mobiilipelimarkkina kasvanut rajusti viime vuosien aikana, ja tämän vaikutuksena indiepelit ovat myös saaneet paljon huomiota. (Indie Games 2016.)

Vaikka yksittäisien pelien suosiota on vaikea ennustaa, ovat nämä markkinat vielä niin kovassa nousussa, että mobiilipelin luonti on aloittelevien pelien kehittäjien paras mahdollisuus onnistua ja mikään ei estä pelien julkaisua muillakin laitteilla. On myös olemassa useita pelimoottoreita, jotka luovat suoraan mobiililaitteille sopivaa koodia, joka helpottaa asioita huomattavasti.

Tässä lopputyössä käyn läpi Kingdom Keeper –nimisen pelin graafisen osuuden toteuttamisen. Kingdom Keeper on minun ja kaverini itse kehittämä peli-idea, jota teemme omalla ajallamme Unity-pelimoottoria käyttäen, pääosin mobiilialustoille. Työnjaon hoitamme niin, että minä keskityn pääosin visuaalisen puolen toteutukseen ja kaverini hoitaa koodauspuolen. Koodausta, tai Unityn syvällisempää käyttöä, ei tässä työssä käydä tarkemmin läpi. Toimimme molemmat pelisuunnittelijoina, joten kuvaan yhteistyötä myös sen tiimoilta.

Visuaalisesta työstä käyn läpi ensin suunnitteluprosessin (luku 4), jossa käyn läpi visuaalisen suunnittelun eri vaiheet ja teknisen työn valmistelun. Siirryn tämän jälkeen työn tekniseen osuuteen, jossa luodaan valmis malli ja käydään läpi enimmäkseen hyviä työtapoja, mielessä pidettäviä seikkoja ja ongelmien välttämistä (luku 5).

## 2 Käsitteet

Asseti – Mikä tahansa 3d malli tai kuva, jota käytetään pelissä.

Konseptikuva – Kuva, joka selventää mitä tahansa ideaa, tai konseptia.

Mappi – Tekstuuri ”kartta” tai kuva

Pelimoottori – Ohjelmistopaketti, joka pyörittää peliä. Esim. Unity.

Pelimekaniikka – Miten peli käytännössä toimii puhtaasti teknisellä tasolla, sisällöstä ja visuaaleista erotettuna.

Sprite – 2d-kuva jota käytetään peli objektina.

Unity – Hyvin suosittu pelimoottori, jota tässä projektissa käytetään.

## 3 Pohja Kingdom Keeper –pelin suunnittelulle

### 3.1 Lähtökohta

Meillä oli toiveena luoda jonkinlainen peli, joka olisi nopea toteuttaa ja saada myyntiin.

Pitkän keskustelun tuloksena päätimme tehdä niin sanotun Tower Defense / RTS (real time strategy) -hybridin. Pelin tavoitteena on selvittää vihollisten hyökkäyksistä, joten tavoitteena on ainoastaan tukikohdan puolustus. Tukikohdan puolustus tehdään pääosin puolustustorneja rakentamalla Tower Defense -peliin tyyliin. Joukkoja voi myös kouluttaa, mutta niillä ei voi hyökätä vihollisten tukikohtaan kuten RTS-peleissä yleensä. Päätimme antaa pelille työnimeksi ”Kingdom Keeper”.

Tämän lopputyön kirjoitusvaiheessa peliprojekti oli hyvinkin varhaisissa vaiheissa ja tavoitteena oli saada valmiiksi ns. prototyyppi, josta projektia voidaan sitten helposti jatkaa. Tässä lopputyössä kuvaan yhden pelissä käytetyn asian, tässä tapauksessa tornin, luontiprosessin suunnittelusta lopulliseen malliin.

### 3.2 2D vai 3D

Alussa valitsimme grafiikan teknisen tyylin eli millä tekniikoilla pelin grafiikat tullaan toteuttamaan. Tämä ei siis liity mitenkään pelin sisältöön, vaan ainoastaan tekniseen toteutukseen. Tässä projektissa grafiikan tekninen tyyli valittiin pääosin sen perusteella, mitä minä pystyn tekemään yksin ilman sen suurempaa ulkopuolista apua. Lähes kaikki suunnittelussa tehdyt valinnat tehtiin tätä silmällä pitäen.

Hahmot tulevat olemaan ”2D-spritejä” eli kuvia, jolla ei ole syvyyttä ja jotka kääntyvät aina kameran suuntaan. 3D-työ tulee olemaan suhteellisen yksinkertaista, jossa malleihin ei tule paljon yksityiskohtia ja tekstuurit ovat pieniä ja pikselimäisiä. Näin pystyn tuottamaan mahdollisimman nopeasti ja helposti erilaisia ”asetteja”, joita tullaan tarvitsemaan melkoinen määrä. Assetilla tarkoitetaan niitä rakennuspalikoita, joista pelissä olevat asiat tai yksittäiset objektit koostuvat. Näitä voivat olla esimerkiksi 2D-kuvat, 3D-mallit ja tekstuurit.

Aluksi ideani oli tehdä peli täysin 2D:nä, mutta ohjelmoijan mielestä 3D on ohjelmointipuolella helpompi toteuttaa. Esimerkiksi fysiikkalaskentaa, josta peli on hyvin riippuvainen, on paljon helpompi tuottaa 3D-tilassa kuin 2D-tilassa. 3D-työ vie kuitenkin enemmän aikaa kuin 2D-työ, joten päädyimme käyttämään molempia tekniikoita. 3D-tilaan saa myös hienompia valoeffektejä, joilla voidaan luoda tunnelmaa ja tuoda kuvaan eloa.

Nämä tekniset asiat määrittävät jo pitkälti pelin esteettistä ilmettä (Kuva 1). Täyttä realismia ei tulla hakemaan sen monimutkaisuuden vuoksi, vaan päädytään jonkinlaiseen piirretympään yleisilmeeseen. Tällaiseen yksinkertaiseen ulkonäköön sopivat kirkkaat värit ja pelkistetyt tekstuurit. Realistiset tekstuurit toisivat vain ylimääräistä sekasortoa ruutuun, tai ainakin realistisen ulkoasun hyväksi hiominen veisi paljon enemmän aikaa. Kaikin puolin pikselimäinen ulkonäkö on myös miellyttävän näköistä, trendikkään retroa ja tuo mieleen vanhempia pelejä luoden nostalgiaa. (Gnomon School 2014.) On mielenkiintoista yhdistää 2D-ilmettä uuteen 3D-ajan toteutukseen.



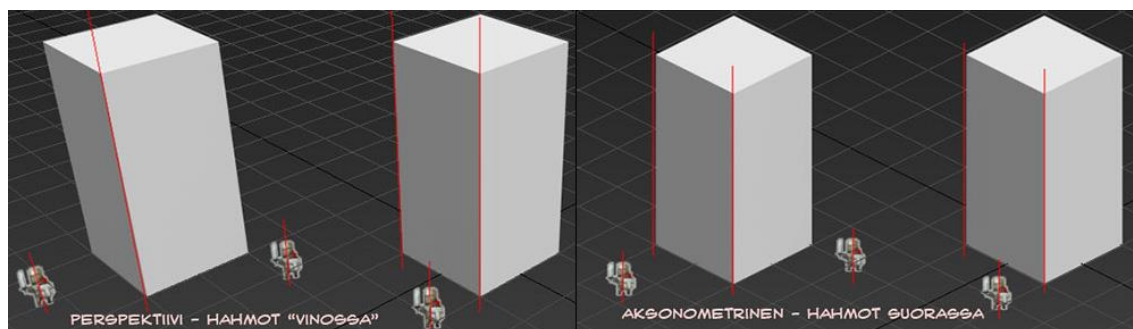


## Kuva 1. Realistinen vai piirrettyinä?

### 3.3 Kamera

Oikeanlaisen kameran valinta on asia, jota emme ole vielä loppuun asti miettineet. Huomasimme muutamaa testiä tehdessämme, että 2D-spriteit näyttävät oudoilta ruudun laidoilla, jos käytössä on normaali perspektiivikamera. Ne ikään kuin seisovat viinossa. Tämän ongelman korjaa kameran vaihtaminen aksonometriseen kameraan, mutta siinä on taas omat huonot puolensa.

Aksonometrinen kamera tuottaa kuvan eri tavalla kuin perspektiivikamera; se käyttää aksonometristä projisointia. Aksonometrisessä projisoinnissa objektit eivät pienene horisonttiin päin, vaan kaikki objektit näkyvät ruudulla ikään kuin ne olisivat samalla tasolla. Tämän lisäksi kaikki vertikaaliset viivat ovat täysin samansuuntaisia. Tämä tarkoittaa myös sitä, että 2D-spriteit "seisovat suorassa" missä tahansa kohtaa ruutua ne sijaitsevatkin. Yleisesti aksonometrinen projisointi näyttää hieman oudolta, mutta on kyllä aivan toimiva ja yleisesti käytetty vaihtoehto pelimaailmassa. (Koncewicz 2009.)



Kuva 2. Perspektiivi vai aksonometrinen kamera?

Aksonometrinen kamera korjaa spritejen oudon kääntäilyyn ruudun reunoilla, mutta se myös tarkoittaa sitä, että 3D-syvyysilluusio kärsii hieman (Kuva 2). Emme ole vielä päättäneet, kumpaa ratkaisua lopullisesti käytetään, mutta se selviää projektin edetessä kun pääsemme tekemään enemmän testejä.

Mahdollisesti hyväksymme spritejen vääristymisen ja päädymme kuitenkin perspektiivikameraan syvyysilluusion vuoksi ja koska se antaa meille enemmän vapautta kamerakulmien kanssa. Perspektiivikameraa pystyy esimerkiksi helpommin kääntämään ylös tai alas, kun taas ortografisen kameran täytyy olla lukittuna tiettyyn kulmaan.



### 3.4 Vastaavien toteutuksien tutkiminen

Tutkimme hieman myös muita pelitoteutuksia, joissa on käytetty samanlaisia tekniikoita ja tyylejä. Etsimme näistä ideoita ja analysoimme, mikä niissä toimii hyvin ja mitä voisi tehdä paremmin. Samoihin ongelmiin, joihin me törmäämme, ovat toiset mahdollisesti jo törmänneet, joten ratkaisuja saattaa olla jo olemassa. Kaikessa ei tarvitse keksiä pyörää uudelleen.

Otin tähän kolme esimerkkiä peleistä, jotka eniten inspiroivat peliprojektiamme. Näissä peleissä on hyviä esimerkkiratkaisuja, joista voimme ottaa oppia. Poimimme asioita, joista pidimme ja kehitimme oman tyylimme niitä hyödyntäen.

Ensimmäisenä esimerkkinä on puhtaasti 2D-tekniikalla toteutettu Kingdom Rush –peli (Kuva 3). Meidän pelissämme olevat hahmot ovat hyvin samanlaisia tämän pelin 2D-grafiikalla tehtyjen hahmojen kanssa. Kingdom Rushissa näimme myös, miten 2D-maailman fysiikat eivät aina toimi oletetulla tavalla. Esimerkiksi nuolten lentoradat eivät aina käyttäydy luonnollisesti, kun taas 3D-maailmassa ne on helpompi toteuttaa.



Kuva 3 Kingdom Rush.

Toisena esimerkkinä toimii Warcraft 3 –peli (Kuva 4), jossa on hyvin samanlaisia elementtejä kuin edellä mainitussa Kingdom Rushissa, mutta Warcraft 3 on toteutettu täysin 3D-tekniikalla. Tästä saamme hyvää osviittaa siitä, miltä rakennukset voisivat näyttää. Emme kuitenkaan halua tehdä hahmoja 3D:nä sen suuren työn vuoksi, eikä se sovi meidän piirretympään visioon. Tämänkään kokoisia 3D-malleja ei näe ruudulta

kovin helposti, ja meidän hahmoistamme tulee tätäkin pienempiä, joten toteutamme hahmot 2D-spriteillä.



Kuva 4. Warcraft 3.

Kolmantena inspiraation lähteenä toimii Return Fire –peli (Kuva 5). Tämä peli on hyvin lähellä meidän valitsemaamme teknistä toteutusta, koska maailma on 3D:tä ja hahmot, puut ja muut propsit ovat 2D-spritejä. Tämä antoi meille suurimman luottamuksen siitä, että tämän tyylinen tekninen toteutus voi toimia ja sillä voi saada pelille mieluisan ulkonäön.



Kuva 5. Return Fire.

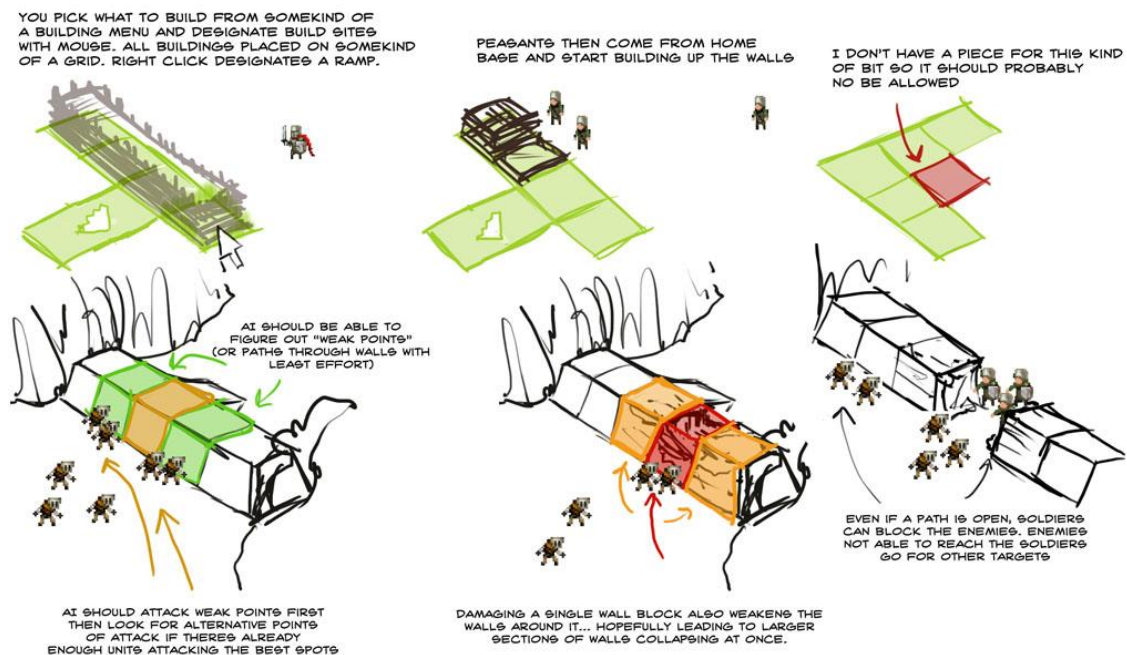


## 4 Pelin grafiikan suunnitteluprosessi

### 4.1 Lähtökohta

Tämän prosessin aikana keskustelimme paljon koodaajan kanssa pelin eri mekaniikoista ja yritimme lyödä lukkoon pelin vision. Tavoitteena oli luoda selvä kuva siitä, mitä peli tulee sisältämään ja miltä se tulee näyttämään sekä valmistautua pelin tekniseen toteutukseen.

Artistin näkökulmasta tämä tarkoittaa käytännössä suuren määrän konseptikuvien luontia ja visuaalisen ilmeen hakemista. Pelisuunnitteluosuutta helpottamaan loin myös puhtaasti kommunikaatiovälineiksi tarkoitettuja kuvia, joilla selvennettiin esimerkiksi pelimekaniikkaa eli miten peli toimii (Kuva 6).



Kuva 6. Esimerkki pelimekaniikkojen suunnittelusta.

Tässä vaiheessa selvillä on pelkästään pelin perusidea: halutaan luoda peli, jossa puolustetaan tukikohtaa ulkopuolisia vastaan. Tällaisen perusidean voisi upottaa melkein mihin maailmaan tahansa, ja sen logiikka pysyy samana. Ritarit voivat puolustaa linnoitusta, ihmiset voivat puolustaa avaruusasemaa avaruusolentoja vastaan tai vihreät siat voivat puolustautua vihaisten lintujen hyökkäyksiä vastaan. Täytyy siis päättää pelille pelimaailma eli ympäristö ja hahmot, joilla pelataan.

Tähän peliin valittiin hyvin perinteinen keskiaikainen fantasiamaailma, joka sopii hyvin peliin, jossa puolustetaan tukikohtaa. Linnat ja jousipyssyt, miekat ja taikuus toimivat hyvin osana peliin valittua fantasiateemaa ja pelilogiikkaa. Tästä tyylistä löytyy myös suuri määrä inspiraatiolähteitä elokuvista, peleistä, kirjoista ja oikean maailman historiasta.

Vaikka perinteinen keskiajan fantasiamaailma on usein käytetty aihe, on sen käyttämisessä myös tiettyjä etuja. Pelaaja tunnistaa käytetyn teeman ja ymmärtää näin helpommin, mistä pelissä on kyse. Jos luodaan liian vieras maailma, pelin logiikka ei välttämättä aukea yhdellä silmäyksellä ja peli tuntuu sekavalta. (Edlin 2015.) Maailmasta voi kuitenkin tehdä kiinnostavan pienilläkin asioilla, ja paljon käytettyyn teemaan voi tuoda oman näkökulmansa.

#### 4.2 Ideoiden haku, referenssien keräys, moodboard

Suunnitteluprosessi on hyvä aloittaa keräämällä mahdollisimman paljon referenssejä ja ideoita, joista voi hakea inspiraatiota työn eri vaiheissa. Referenssejä haetaan toki myös työn edetessä, mutta olen huomannut, että työ on hyvä aloittaa puhtaasti referenssimateriaalin keräämisellä ennen kuin aloittaa varsinaisen työn.

Keräsin tätä projektia varten suuren määrän kuvia ritareista ja linnoista, erinäköisistä keskiajan rakennelmista ja teknologiasta. Keräsin muiden artistien maalauksia ja luonnoksia eri lähteistä. Myös elokuvista ja muista peleistä löytää paljon hyödynnettävää materiaalia. Tarkoituksena ei tietenkään ole suoraan kopioida näistä mitään, vaan näiden kuvien ideana on antaa inspiraatiota ja mahdollisia ideoita. Oikean maailman valokuvista saa tuotua fantasiamaailmaan realistisia elementtejä, joista voi ottaa mallia omiin luonnoksiin, ja ne antavat fantasiamaailmalle todenmukaisuutta. Erilaisia elementtejä sekoittamalla voi luoda pelille oman todellisuutensa. (Edlin 2015.)

Tässä vaiheessa on hyvä myös kerätä niin sanottuja ”moodboardoja” eli kuvakokoelmia, joilla määritellään pelin yleistunnelmaa tai visiota (Kuva 7). Niitä voi kerätä henkilökohtaiseen käyttöön tai ryhmän välisen kommunikaation työvälineeksi. Ne auttavat havainnollistamaan, minkälaista tunnelmaa ja visuaalista ilmettä ollaan tavoittelemassa.



Kuva 7. Moodboard

#### 4.3 Yleiskuvien teko

Aloitan työni niin sanotuilla yleiskuvilla (Kuva 8). Näiden maalausten tarkoituksena on rakentaa ideoista kokonaisuuksia ja tarkentaa kuvaa maailmasta, ideoista ja mahdollisuuksista menemättä liikaa yksityiskohtiin.

Haen kuviin yleistä ilmettä ja tunnelmaa. Valitsen kuviin aiheita, jotka tukevat pelin keskeistä teemaa, kuten kuvia sotilaista taistelemassa vihollista vastaan. Näin kuka tahnansa kuvan näkevä tietää, mistä pelissä on kyse: ritareita puolustamassa linnoitusta.



Kuva 8. Yleiskuva

Tässä vaiheessa en vielä erityisemmin välitä yksityiskohdista tai lopullisista konsepteista. Kerään vain yleisiä asioita, joista voidaan sitten poimia ideoita sekä elementtejä myöhempää suunnittelua varten. Nämä kuvat voi tehdä miten tarkasti haluaa tarpeen mukaan. Minä teen yleensä tässä vaiheessa hyvin luonnosmaisia kuvia, mutta jos ne veisi pitemmälle, niitä voisi käyttää jopa promo-materiaalina. Tämän voi tehdä myöhemminkin pelin mainostusvaiheessa; viimeistellä jonkin yleiskuvista vaikkapa julisteeksi.

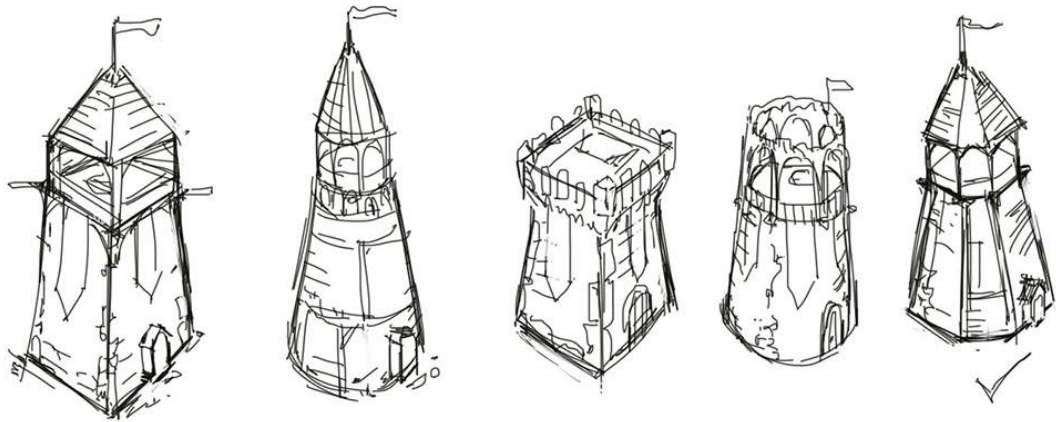
#### 4.4 Ideoiden hiominen ja yksittäiset konseptit

Kun pelin yleisilme on saatu luotua, on aika valita asioita, jotka tarvitsevat tarkempaa suunnittelua. Tarkoitus on karsia ideoita ja kiteyttää ne asiat, jotka varmasti tullaan tarvitsemaan pelissä. Ideoita mietitään pidemmälle ja luodaan yksittäisiä selvempiä kuvia.

Esimerkiksi tässä peliprojektissa tarvitaan torni. Yleiskuvissa piirsin jo muutaman tornin, mutta aloin silti piirtämään erilaisia torneja käyttäen referenssikansiota apuna ideoiden luomiseen. Vaikka olisikin jo jokin idea mielessä, kannattaa aina tutkia aiheita hieman laajemmin kokeilemalla erilaisia muotoja ja materiaaleja. Piirtäessäni pidin mielessä, miten torni sopii pelimaailmaan ja voisiko siihen keksiä jotain uniikkia. Torneja



suunnitellessani pyrin miettimään, miten tornit olisi rakennettu siihen aikaan ja mitä materiaaleja heillä olisi ollut käytettävänä.



Kuva 9. Erilaisia torneja

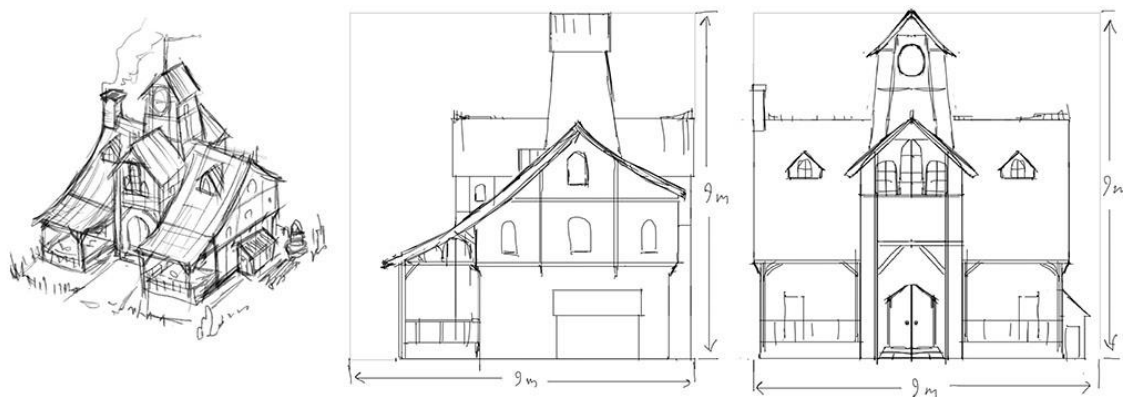
Pidän myös mielessä, että objektit ovat helposti tunnistettavia ja erottuvat toisistaan (Kuva 10). Objektien siluetit ja värivalinnat auttavat pelaajaa tunnistamaan ruudulla olevat objektit nopeasti ilman sekaannusta. (Valve 2007.) Esimerkiksi erilaisia värejä ja koristeita käyttämällä voin luoda kaikille pelaajan rakennuksille yhteisen teeman, joka luo yhtenäisyyttä ja auttaa pelaajaa näkemään helposti omat joukkonsa vihollisten seasta. Tätä kutsutaan yleisesti luettavuudeksi.



Kuva 10. Siluetit, värit ja luettavuus.



Lopuksi luon tarvittaessa yksittäisille mallinnettaville asioille tarkemmat kaavakuvat (Kuva 11.) Näistä kuvista pitäisi löytyä kaikki, mitä lopulliseen malliin tarvitaan. Teen kuvat pääosin edestä ja sivusta päin, ja lisään muita kuvakulmia tarpeen mukaan. Suurinta osaa pelin rakennuksista ei tulla näkemään kuin yhdestä suunnasta pelikameran takia, joten rakennuksen takaosaa ei tarvitse sen tarkemmin suunnitella.



Kuva 11. Pelissä käytetyn talon kaavakuva.

Tällaisten kaavakuvien teko tulee erityisesti tarpeeseen, jos mallinnustyötä tehdään toisen artistin kanssa. Kaavakuva takaa sen, että mallista tulee konseptiartistin vision mukainen toteuttajasta riippumatta. Se on myös suuri apu mallintajalle, koska siitä saa tarkempaa tietoa kuin pelkästä konseptikuvasta, jolloin mallintajan ei tarvitse arvuutella yksityiskohtia.

#### 4.5 Mockup ja user interface

Mockup on kuva, jonka on tarkoitus antaa kuva pelin lopullisesta ulkonäöstä. Mockupit ovat niin sanottuja "vale-screenshoteja", joista nähdään, miltä peli ehkä tulisi näyttämään, kun peliä pelataan. Niissä näkyy varsinaisen pelin lisäksi myös pelin käyttöliittymää (Kuva 12.)

Nämä kuvat voivat alkuvaiheessa olla hyvinkin yksinkertaistettuja ja nopeasti tehtyjä, koska niiden päätarkoitus on tiimin välinen kommunikaatio ja ideoiden selventäminen (Anhut 2014). Pelin ja sen käyttöliittymän lopullinen ulkonäkö tulee varmasti vielä muuttumaan projektin etenemisen aikana, joten tässä vaiheessa siihen ei kannata uhrata turhan paljon aikaa.



Kuva 12. Mockup

Käyttöliittymästä ("user interface", UI) suunnitellaan ensimmäinen versio pitäen mielessä miten peliä pelataan, mitä informaatiota pelaajalle täytyy sen kautta kommunikoida ja mitä, mahdollisia nappeja pelaaja tarvitsee peliä pelataksaan. Kingdom Keeper -pelissä tarvitaan esimerkiksi pienoiskartta eli "minimap", resurssiluvut kuten esimerkiksi rakennustarvikkeiden määrä, pelaajan ja joukkojen voimat sekä rakennusmenu josta valitaan, mitä rakennetaan. Mietin myös mahdollisia ilmoituksia, joita pelaaja saattaa tarvita pelin aikana, että pelaaja ymmärtää pelin kulkua. Käyttöliittymää suunnitellessa on hyvä muistaa, että se on myös suuri osa pelin visuaalista ilmettä ja luo pelille omaa luonnetta. Se kannattaa siis tehdä teemaan sopivaksi. (Oppenlander 2015.)

## 5 Grafiikan tekninen toteutus

### 5.1 Teknisen työn aloitus

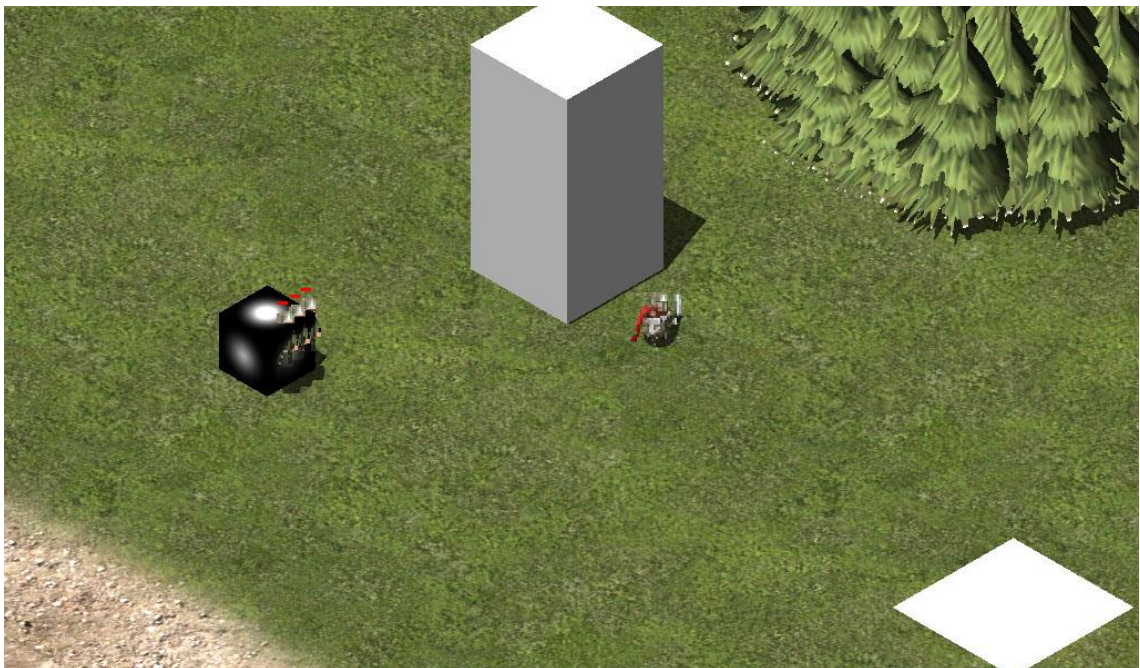
Kun pelin visuaalinen suunnittelu on saatu hyvälle mallille, aloitetaan teknisen osuuden suunnittelu ja toteutus. Ideoita lyödään lukkoon ja aletaan työstää niitä valmiiksi asetteiksi. Suunnitteluosuus saattaa toki jatkua vielä tässäkin vaiheessa, mutta yleinen kuva siitä, mitä ollaan tekemässä, pitäisi olla koossa.

Tähän peliprojektiin teen mallinnuksen 3Ds Max -ohjelmalla ja tekstuurit luon Photoshopia käyttäen. Käyn Maxista ja Photoshopista läpi asetuksia ja ominaisuuksia, joista osa saattaa olla ohjelmakohtaisia, mutta yleisesti asioita voi soveltaa muissakin ohjelmissa. Näistä voisi mainita esimerkkeinä 3D-ohjelmat Maya ja Blender sekä kuvankäsittelyohjelman GIMP.

Tässä osuudessa käyn läpi pääosin hyviä työtapoja ja asioita, mitä kannattaa huomioida työn aikana. Esitän myös muutamia ongelmia, joihin törmäsimme, ja annan näihin ratkaisuehdotuksia. Tavoitteena on luoda mahdollisimman hyvälaatuisia malleja ja aseteja, jotka ohjelmoijan on helppo laittaa paikalleen.

Tärkeä asia on ohjelmoijan tai muun tiimin kanssa keskustelu suunnitelmista ja lopullisten konseptien hyväksyminen. Suunnittelun aikana on jo saatettu käydä keskustelua siitä, mitä suunnilleen tullaan tekemään, mutta viimeistään tässä vaiheessa suunnitelmat kannattaa tuoda koko tiimin tietoon. Tässä kommunikoinnissa voi hyödyntää esimerkiksi apukuvia.

Aloitimme ohjelmointityön luomalla pelistä aluksi hyvin yksinkertaisia testiversioita eli niin sanottuja ”palikkaversioita”, joissa olemme kokeilleet muutamia ideoita ja vetäneet johtopäätöksiä siitä, miten peli tulee toimimaan (Kuva 13). Teimme myös kokeiluja erilaisilla mittasuhteilla, koska ne vaikuttavat suoraan mallinnustyöhön.



Kuva 13. Pelin varhainen testiversio. ”Palikkaversio.”

Tulimme siihen lopputulokseen, että piirtämämme 12 pikseliä korkea hahmo on pelimaailmassa 2 metrin pituinen. Tähän mittaan vertaamalla päätimme pelin muut mitayksiköt. Esimerkiksi muurin pohja tai ”jalanjälki” on 3x3 metriä ja sen korkeus on 4 metriä. Tämän perusteella päätimme, että pelin rakennukset käyttävät 3x3-metristä gridiä, eli kaikki rakennukset ovat tämän mitan kerrannaisia. Majatalon pohja on esimerkiksi 9x9 metriä. Muureja pienemmät ns. koristeellisemmat mallit voimme kuitenkin asetella vapaasti minne haluamme. Myös pelihahmot voivat liikkua vapaasti pelimaailmassa ilman gridin rajoitusta. Tällaisten mittojen määrittäminen auttaa tulevaisuudessa asetejen luonnissa ja pitää asiat yhteensopivina.

Näiden päätösten perusteella apukuvia apuna käyttäen, ohjelmoija voi alkaa luomaan tarkempaa prototyyppiä pelistä käyttäen yksinkertaisia palikoita valmiiden mallien sijasta. Sitä mukaa kun malleja valmistuu, ne voidaan vaihtaa palikoiden tilalle, jolloin niitä voi kokeilla toiminnassa heti.

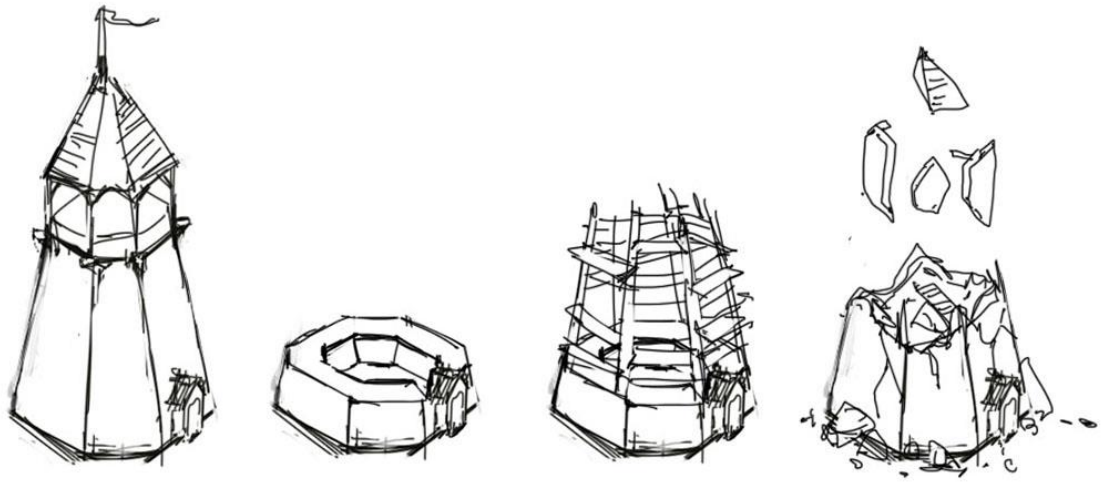
## 5.2 Assetien luonti

Nyt kun ohjelmoija on saatu töihin, graafikko alkaa luoda aseteja. Tässä projektissa päätettiin luoda ihan ensimmäiseksi malli perustornille. Tämän tornin pohjalta voidaan myöhemmin luoda muita torneja.

Muutaman palikkaversiokokeilun perusteella päädyimme torniin, jonka pohja (tai jalanjälki) sopii 6x6-metriseen ruutuun, ja tornin korkeudeksi tuli 12 metriä. Taso, jolla sotilaat tulevat seisomaan, tulee olemaan 8 metrin korkeudessa. Tason korkeudeksi päätettiin 8 metriä, koska perusmuuri on 4 metriä korkea ja haluttiin, että sotilaat pystyvät ampumaan tornista muurin ylitse.

Jokaisessa pelin rakennuksessa on erillinen pohjakerros, joka tulee näkyviin, jos pelaaja kävelee rakennuksen taakse (Kuva 14). Tätä samaa pohjakerrosvaihetta hyödynnetään myös kuvastamaan rakennusvaihetta. Jokaisesta rakennuksesta luodaan myös tuhoutunut versio ja mahdollisia rakennuksesta irrotettavia palasia. Osittaista tuhoutumista esitetään enimmäkseen tekstuurin vaihdolla.

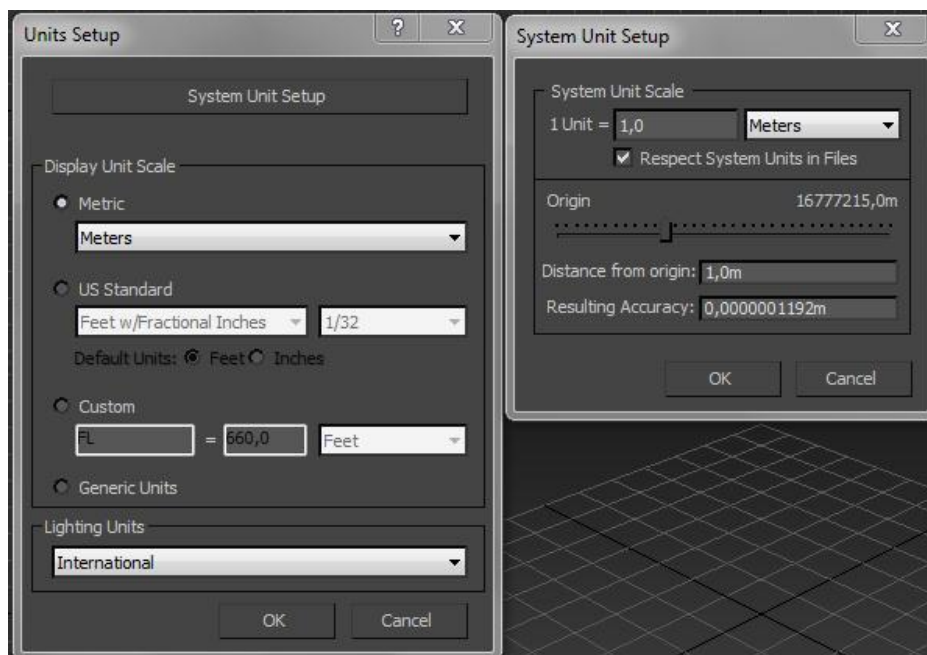




Kuva 14. Torniin tarvittavat erilliset mallit.

### 5.3 Mallinnus

Ennen mallinnuksen aloittamista on erittäin tärkeää tarkistaa, että 3D scene, eli mallinnusohjelmassa auki oleva 3D tila, käyttää oikeita mittayksiköitä. Tämä tapahtuu Maxisissa *Customize* -> *Unit Setup* -valikosta *System Unit Setup* -nappia painamalla (Kuva 15). Mittayksiköiksi valitsin metrit, jolloin 1 Maxin mittayksikkö vastaa yhtä metriä. Nämä luvut siirtyvät suoraan Unityyn exportausvaiheessa, jolloin objekteja ei tarvitse erikseen skaalata jälkikäteen.

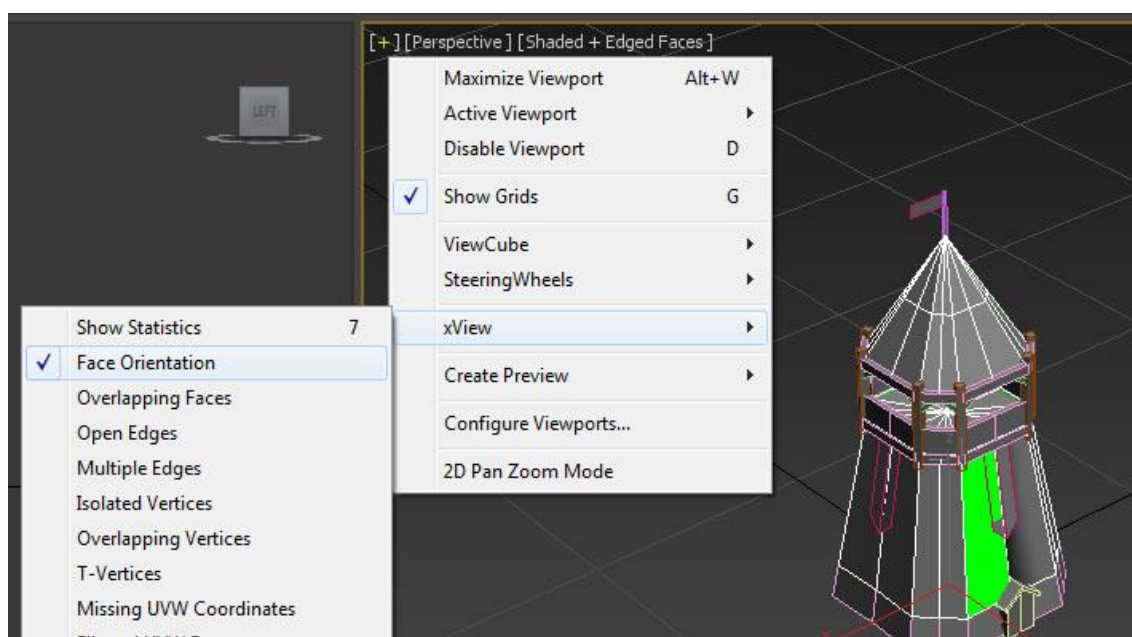


Kuva 15. Yksiköiden asettelu.

Mallit pidetään mahdollisimman yksinkertaisina ja siisteinä. En tuhlaa aikaa turhien yksityiskohtien mallintamiseen, koska se veisi liikaa aikaa eikä sopisi pelin yksinkertaistettuun yleiskuvaan. Poistan myös malleista kaikki osat, joita pelaaja ei tule koskaan näkemään. Tällaisia ovat esimerkiksi mallien sisällä olevat rakenteet tai mallien pohjat. Tämä myös pitää polygonimäärät pieninä, jolloin ruudulle saa enemmän tavaraa suoritustehon kärsimättä. (Silverman 2013.)

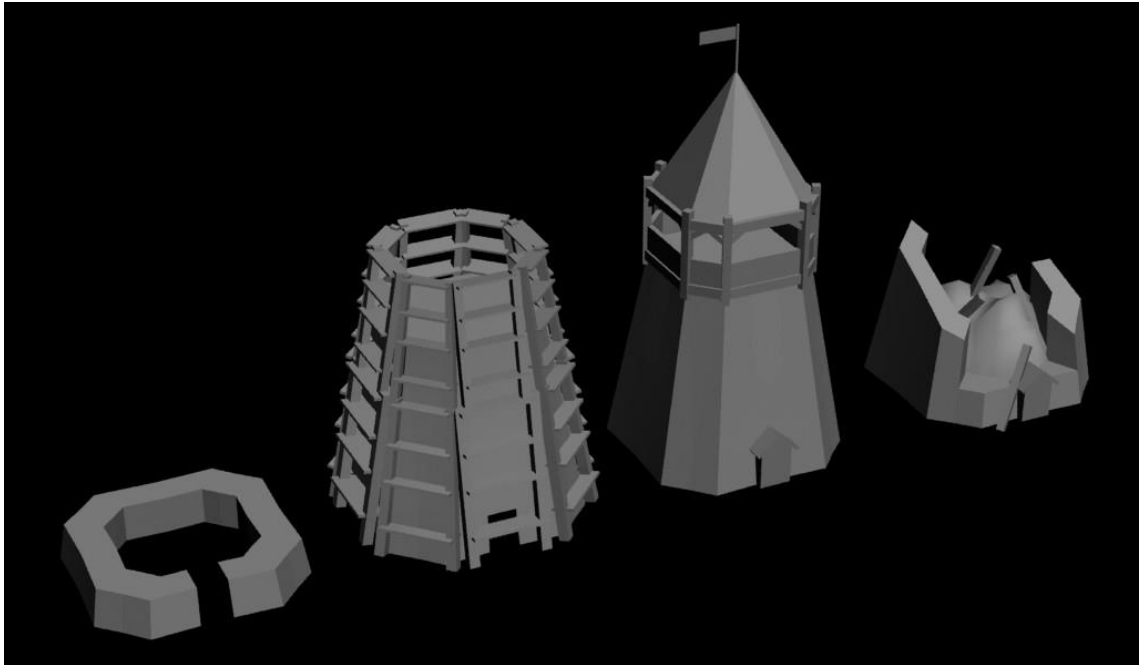
Mallinnuksessa kunnioitetaan neljän kulman sääntöä, jolloin tavoitteena on, että jokaisella pinnan osalla eli "facella" on maksimissaan neljä kulmaa. Exportausvaiheessa ohjelma jakaa mallin automaattisesti kolmioihin. Jos facella on tällöin enemmän kuin 4 kulmaa, joutuu ohjelma tekemään arvauksen, miten face kuuluu jakaa, jolloin lopputulos saattaa olla odottamaton ja joskus ongelmallinen. Kun pitäydytään neljän kulman säännössä, ei odottamattomia ongelmia pitäisi löytyä. Tämä on yleisesti käytössä oleva paras työtap, joka luo ns. "puhtaan mallin". (Silverman 2013.)

Mallintaessa huomasin, että 3D Maxin mirrorointityökalua kannattaa välttää kokonaan. Tämä työkalu peilaa mallit skaalaamalla ne lukuun -1. Käytännössä tämä tarkoittaa myös sitä, että polygonien normaalit ovat väärinpäin. Maxissa tätä ei aina näe, mutta kun mallit vie Unityyn, väärinpäin olevat polygonit muuttuvat läpinäkyviksi. Näitä väärinpäin olevia normaaleja on helppo metsästä Maxissa käyttämällä viewportin xview -valikosta löytyvää *Face Orientation* -toimintoa (Kuva 16).



Kuva 16. xView näyttää normaalien suunnan.

Tähän projektiin tarvitsin kolme eri versiota mallista: normaali versio, rakenteilla oleva versio ja tuhoutunut versio (Kuva 17). Tein ensimmäisenä normaalin version, jonka pohjalta loin muut.



Kuva 17. Lopulliset mallit.

#### 5.4 Teksturointi

Tekstuurit luon Photoshop-kuvankäsittelyohjelmalla. Käytän suhteellisen pieniä tekstureita, joita on helppo tehdä ja ne tukevat pelin yksinkertaistettua ulkonäköä. Suurin tekstuuri, jota käytän, on 512x512 pikseliä. Ensimmäiseen mallinnettavaan torniin teen 256x256 pikselin tekstuurin. Koska mallit ovat niin yksinkertaisia, teen tekstuurit ensin ja UV-mapin sen jälkeen.

Aloitan työni hyvin nopealla versiolla tekstuurista, jota sitten kokeilen mallissa (Kuva 18). Yritän löytää paikkoja, joissa voisin toistaa tekstuurin jotain osaa, jolloin jokaista kohtaa ei tarvitse erikseen piirtää tekstuuriin. Etsin myös ongelmakohtia ja paikkoja, joihin voisi lisätä enemmän tarkkuutta. Yritän yleisesti pitää pikselitiheyttä jokseenkin yhtenäisenä.





Kuva 18. Tekstuurien asettelua. Huomioi hyvin tilapäiset tekstuurit.

Teen tekstuureista suhteellisen maalausmaisia, ja koetan välttää turhaa sekasortoa. Pienessä koossa turhat yksityiskohdat vain sekoittavat yleistä ilmettä ja pilaavat grafiikan luettavuutta. Koska peli ei tule omaamaan erityisen korkeatasoista valomootoria, maalaan varjot suoraan tekstuuriin. Pidän varjot omalla layerillaan, jotta voin piilottaa ne esimerkiksi specular-mappia luodessani.

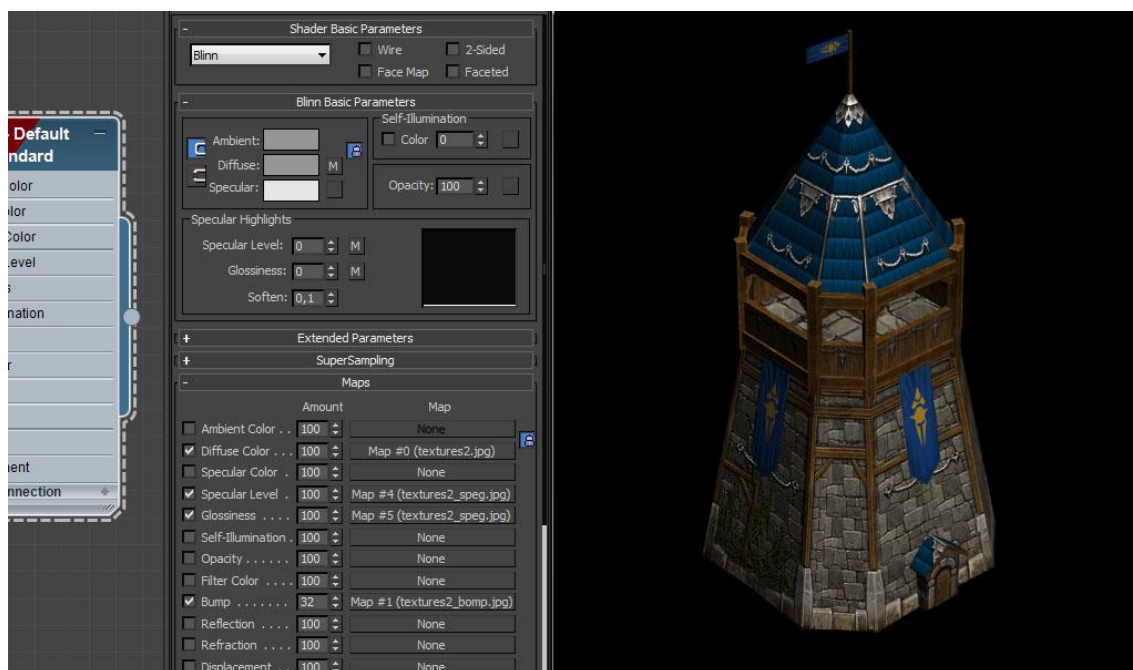
Pelin materiaaleihin tulee käyttöön kolme mappia: diffuse väreille, specular kiiltävyydelle ja normal pinnan yksityiskohdille. Unity osaa käyttää png-tiedostojen alpha-kanavaa, joten erillistä alpha mappia läpinäkyvyydelle ei tarvita. Mahdollisesti tulevaisuudessa tullaan käyttämään specular color mappia heijastusten väritykselle, mutta en tee sitä tässä vaiheessa.

Specular ja normal mapit teen valmista diffuse mappia muokkaamalla. Specular mappi on vain mustavalkoinen versio diffuse mapista, jossa korostan kiiltäviä pintoja. Mitä valkoisempi pinta on, sitä enemmän se heijastaa ympäristöä. Specular color mapilla saadaan heijastus värjättyä eriväriseksi, jolloin voidaan luoda eri materiaalien illuusiota. Jos halutaan esimerkiksi kultaa, tulisi specular color olla kullan ruskea. Teen mustavalkoisen bump mapin pinnan korkeuseroista. Bump mapissa valkoinen tarkoittaa, että pinta kohoaa normaalitasosta, jolloin voidaan tehdä pieniä pinnan yksityiskohtia.



Kuva 19. Valmiit tekstuurimapid: Diffuse, Specular ja Bump

Rakennan myös materiaalit Maxissa (Kuva 20). Maxin oletusmateriaalin voi viedä suoraan Unityyn, kunhan siitä ei rakenna turhan monimutkaista. Liitän luomani mapit oikeille paikoille ja testaan yksinkertaisen valon kanssa, miltä valmis torni tulee näyttämään.



Kuva 20. Materiaaliasetukset ja valmis torni.

## 5.5 Hahmot

Hahmot tässä projektissa ovat animoituja 2D-spritejä, eli kuvia, joilla ei ole syvyyttä. Koska peli toimii 3D-maailmassa, täytyy hahmojen kuitenkin täyttää jokin tila, että peli pystyy laskemaan, missä hahmot ovat ja osuvatko esimerkiksi nuolet niihin. Tätä varten hahmoilla on pelissä näkymätön 3D-laatikko tai "hitbox", joka luodaan Unityssä myöhemmin.

Hahmoilla on neljä kävelysuuntaa, joilla on jokaisella omat animaationsa. Suunnat ja kaantuvat niin, että alaspäin ja oikealle kuljettaessa käytetään yhtä animaatiota ja ylös oikealle kuljettaessa käytetään toista animaatiota. Vasemmalle kulkevat ovat tällä hetkellä vain oikealle suuntautuvan animaation peilikuvat. Mahdollisesti tulevaisuudessa teen niille omat animaatiot, koska peilikuva-animaatiossa hahmo esimerkiksi vaihtaa miekkakättään aina kääntyessään ja tätä en välttämättä halua.



Kuva 21. Esimerkkejä hahmon animaatiosta.

Spritet luon PNG-formaattiin, jonka alpha-kanavaa eli läpinäkyvyyttä voidaan käyttää suoraan Unityssä. Hahmon animaatoruudut laitetaan ns. spritesheet-kuvaan, eli asetelen ruudut 16x16 pikselin gridille. Tästä luon sitten animaation Unityn puolella. Eri ruutuja tuli päähenkilölle yhteensä 17: 6 idle-animaatioon, 8 kävelyyn, 1 lyöntiin, 1 kun hahmoon sattuu ja yksi kuolemaan.

## 6 Assetien viimeistely ja testaus

### 6.1 Export ja Import

Artistin viimeinen työtehtävä on tarkistaa assetit pelimoottorissa. Tässä projektissa Unity -pelimoottorissa. Käyn nopeasti läpi exportauksen ja mallin tuomiseen Unityyn sekä muita asioita, joihin artisti tulee törmäämään. Vaikka käytämme Unitya, samat asiat tulevat vastaan eri pelimoottoreissa, esimerkiksi Ogre 3D:ssä.

Unityn puolella on ohjelmoija jo luonut Unity-projektin ja työstänyt sitä eteenpäin itsenäisesti. Projektista löytyy jo ohjelmoijan luoma torniobjekti, johon malli vain liitetään. Pitää siis varmistaa, että malli toimii oikein ja ongelmia ei ilmaannu.

Unity osaa lukea yleistä FBX-tiedostoformaattia, jota käytetään digitaaliseen tallentamiseen, joten exporttaus on Maxista suhteellisen suoraviivaista. Koitan kuitenkin pitää tiedostot suhteellisen ”puhtaina”, joten export-valikosta otan ruksit pois kaikesta, mitä

tiedostossa ei tarvitse pitää mukana. Otan tiedostoon mukaan vain geometrian ja materiaalit ja mahdollisesti animaatiot, jos niitä on tehnyt Maxissa.

FBX-tiedoston tallennan Unity-projektin *Assets / Models* -kansioon. Siirrän myös tekstuurit *Assets / Textures* -kansioon ja spritet *Assets / Sprites* -kansioon. Näin kun Unity-projektissa ladataan assetit uudestaan, pitäisi mallien ja tekstuurien ilmestyä tiedostovalikkoon ja olla valmiina käytettäväksi.

## 6.2 Assetien tarkistus

Kun malli on saatu Unityyn, pitää se tarkistaa, että kaikki on kunnossa. Teen tämän asettamalla mallin pelimaailmaan ja pyörittelen sitä siinä niin, että näen sen joka kulmasta. Jos malli on kunnossa, voin ilmoittaa ohjelmoijalle, että sen voi ottaa käyttöön. Jos malli ei ole kunnossa, palaan Maxin puolelle korjaamaan ongelmat. Ongelmien korjausta on kuvattu luvussa 5.3.

Koska mallit saattavat olla turhan monimutkaisia esimerkiksi fysiikkalaskujen kanssa käytettäväksi, tehdään niille abstraktimpi malli, ns. collision box, jota käytetään pelilogiikassa. Tällä tornilla collision box voi olla vain yksinkertainen laatikko, joka luodaan Unityssä. Monimutkaisemmat collision boxit voi tarvittaessa luoda Maxissa ja tuoda Unityyn omana objektinaan. Unity osaa myös luoda collision boxeja automaattisesti, mutta se ei aina ole käyttökelpoinen.

Määrittelen myös ohjelmoijan avulla tornin eri "vaiheet," eli missä eri muodoissa torni voi olla. Rakennusvaiheessa käytetään eri mallia kuin normaali tornin malli. Kun torni alkaa tuhoutumaan, vaihtuu sen tekstuuri, mikä ilmi antaa tornin huonon kunnon. Lopulta torni vaihdetaan tuhoutuneeseen malliin.

Nyt torni on toimiva osa peliä ja työprosessi tämän yhden assetin osalta on valmis. Tätä prosessia toistamalla luodaan sitten pelin muut assetit

## 7 Yhteenveto

Halusin tässä lopputyössä luoda Kingdom Keeper -pelille yleisesti kattavan taidesuunnitelman. Halusin antaa pelin taiteelle hyvän pohjan ja käydä läpi kaiken, mitä pelin toteutus tulee tarvitsemaan. Projektin aikana sain hyvän käsityksen itsekin, mitä kaikkea tulisi pitää mielessä työn eri vaiheissa ja miten tärkeää on pohtia asioita, jota saattaa pitää itsestäänselvyyksinä.

Alussa onnistuttiin valitsemaan hyvä toteutustapa pelille, mikä on hyvin mahdollista toteuttaa pienellä tiimillä. Eri pelejä tutkimalla löydettiin hyvä ratkaisu simppeleille, mutta kuitenkin hyvän näköiselle piirrettymäiselle taidesuuntaukselle.

Referenssejä keräämällä ja moodboardia hyväksi käyttäen saatiin aikaan hyvän näköisiä yleiskuvia, joitten pohjalta luotiin tarkempia konsepteja, ja tutkittiin miten luoda hyvän näköinen torni. Samalla rakennettiin pientä prototyyppiä pelistä, jonka kanssa testaillemalla saatiin mallien mittasuhteet kohdalleen ja hyvät kaavakuvat piirrettyä.

Mallinnuksen aikana löydettiin hyvät työtavat ja ratkottiin ongelmia. Konseptien pohjalta luotiin puhdas low poly malli joka toimi mainiosti pelissä, joka sitten teksturoitiin kauniisti. Unityn kanssa työskentelyä selvennettiin hieman myös.

Lopputyötä tekiessäni minulle oli pitkään ongelmana aiheen rajaus. Mietin pitkään, pitäisikö minun sisällyttää työhön osio kenttäsuunnittelusta. Maailman luonti pelimoottorissa sivuaa paljon artistin osuutta projektissa, joten se yleisesti ottaen sopisi tähän. Mutta siitäkin ehkä sopivin osuus on tekstuurien luonti, josta kirjoitin jo. Tulin lopulta siihen tulokseen, että aihe on niin laaja, että siitä saisi helposti oman lopputyönsä. Se on kuitenkin asia, mihin itse sukellan seuraavaksi.

Projektin aikana luotiin erittäin hyvän peruspaketti pelin taiteen pohjaksi. Näiden konseptikuvien, mallien ja tekstuurien perusteella ilmenee projektin taidesuuntaus, työtapa ja laatuvaatimukset. Joten, jos niin haluttaisiin, projektin taideosuuden voisi jo jopa luovuttaa toisen artistin vastuulle ja antaa tämän tornin esimerkiksi.

Löydettiin siis hyvin käyttökelpoinen tapa luoda pelille taidesuuntaus ja toimiva suunnitteluprosessi.

## Lähteet

Anhut Anjin 2014. Let's Get Real About Concept Art [verkkodokumentti]  
<http://howtonotsuckatgamedesign.com/2014/02/lets-get-real-concept-art/>  
(luettu: 1.5.2016)

Edlin Tyler 2015. World Building Design [verkkodokumentti]  
<https://selz.com/item/54a45b60b7987212ec103a0a>  
(luettu: 1.5.2016)

Gnomon School 2014. VFX Minimalism: The Beauty of Low-Poly Art. [verkkodokumentti]  
<https://www.gnomon.edu/blog/vfx-minimalism-the-beauty-of-low-poly-art>  
(luettu: 1.5.2016)

Indie Games 2016. The Indie Game Movement [verkkodokumentti].  
<http://www.indiegames.com/what.php>  
(luettu: 1.5.2016)

Koncewicz Radek 2009. A Layman's Guide to Projection in Videogames. [verkkodokumentti]  
<http://www.significant-bits.com/a-laymans-guide-to-projection-in-videogames>  
(luettu: 1.5.2016)

Valve Software 2007. Illustrative Rendering in Team Fortress 2 [verkkodokumentti]  
[http://www.valvesoftware.com/publications/2007/NPAR07\\_IllustrativeRenderingInTeamFortress2.pdf](http://www.valvesoftware.com/publications/2007/NPAR07_IllustrativeRenderingInTeamFortress2.pdf)  
(luettu: 1.5.2016)

Oppenlander Brian 2015. Game UI design [verkkodokumentti]  
[http://www.gamasutra.com/blogs/BrianOppenlander/20151223/262574/Game\\_UI\\_design.php](http://www.gamasutra.com/blogs/BrianOppenlander/20151223/262574/Game_UI_design.php)  
(luettu: 1.5.2016)



Silverman David 2013. 3D Primer for Game Developers: An Overview of 3D Modeling in Games [verkkodokumentti]

<http://gamedevelopment.tutsplus.com/articles/3d-primer-for-game-developers-an-overview-of-3d-modeling-in-games--gamedev-5704>

(luettu: 1.5.2016)

## Kuvalähteet

Kuva 1a, The Creative Assembly, Rome Total War [verkkodokumentti]

<http://www.rocketchainsaw.com.au/wp-content/uploads/2013/06/rome-2a.jpg>

(luettu: 1.5.2015)

Kuva 1b, Blizzard Entertainment, World of Warcraft [verkkodokumentti]

<http://static.quim.co.uk/sys-images/Guardian/About/General/2008/11/13/1226588516776/Gallery-World-of-Warcraft-006.jpg>

(luettu: 1.5.2015)

Kuva 3, Ironhide game studio, Kingdom Rush [verkkodokumentti]

[http://www.macgamestore.com/images\\_screenshots/kingdom-rush-19852.jpg](http://www.macgamestore.com/images_screenshots/kingdom-rush-19852.jpg)

(luettu: 1.5.2015)

Kuva 4, Blizzard Entertainment, Warcraft 3 [verkkodokumentti]

<http://i1-news.softpedia-static.com/images/news2/Blizzard-Releases-All-Warcraft-3-Assets-in-Starcraft-2-472003-6.jpg>

(luettu: 1.5.2015)

Kuva 5, Silent Software, Return Fire [verkkodokumentti]

<http://www.pocketcubes.com/wp-content/uploads/return-fire.jpg>

(luettu: 15.2.2015)